OP1: Naming

I propose to modify DNS to enable an IP packet destined for a particular service to be *piggybacked* over the DNS query for the name of that service, such that upon successful name resolution, a DNS server will send the piggybacked packet to its destination on behalf of the client. With piggybacking DNS (pDNS), a client need not make a separate query to resolve a name and is never aware of any name resolution. I will first describe pDNS under the simplifying assumption that no packet needs fragmentation, and then characterize the performance and scalability of my proposal as compared to the current system.

Under pDNS, a client wishing to make a request to an Internet service constructs a corresponding IP packet while leaving the destination address blank, and encapsulates the packet in the payload of a pDNS query for the name of the service. (Note that the pDNS query is itself contained in an *outer* IP packet.) If a pDNS server that receives such a query is able to directly resolve the name, it fills in the destination address in the encapsulated packet and *dispatches* it to be routed by the underlying IP network. Otherwise, it follows its regular DNS policy to forward the entire query to another server, and waits for a response containing the resolved address so that it may update its cache. In either case, it propagates the response to the source of its own outer packet.

pDNS performance crucially depends on caching. Like traditional DNS, most pDNS queries will be resolved without even exiting the ISP's subnetwork into the public Internet, due to intermediate caching servers, often including one at the client. pDNS would not impose any additional network costs on the route from the client to the ISP's regional servers, which most other Internet requests would have to take anyway. We also note that Internet traffic generally follows a power law distribution in which a few very popular domains, whose resolved addresses can be cached close to the client, account for the majority of requests [1].

As the number and size of requests grows, pDNS imposes higher memory requirements on servers than DNS, but the number and hierarchical structure of servers needed for a given namespace is the same. DNS queries and responses are on the order of 100 B, while encapsulated IP packets can be up to 65 kB, implying pDNS requires at worst 650× more memory. However, as noted above, pDNS queries are rarely expected to travel very far from the client, and local servers generally have more spare capacity than shared servers in high demand. pDNS also retains the load-balancing and anycast routing benefits of DNS.

The main advantage of pDNS is that a client does not wait for a DNS response before sending its actual request, whose route would begin largely identically (until the ISP's regional servers) anyway. Name resolution is also conceptually abstracted away for the client, simplifying higher-level application design.

References

[1] Jeff Desjardins. "The 100 Websites That Rule the Internet." *Visual Capitalist*, 7 Mar. 2017, https://www.visualcapitalist.com/100-websites-rule-internet/. Accessed 17 Oct. 2022. 1